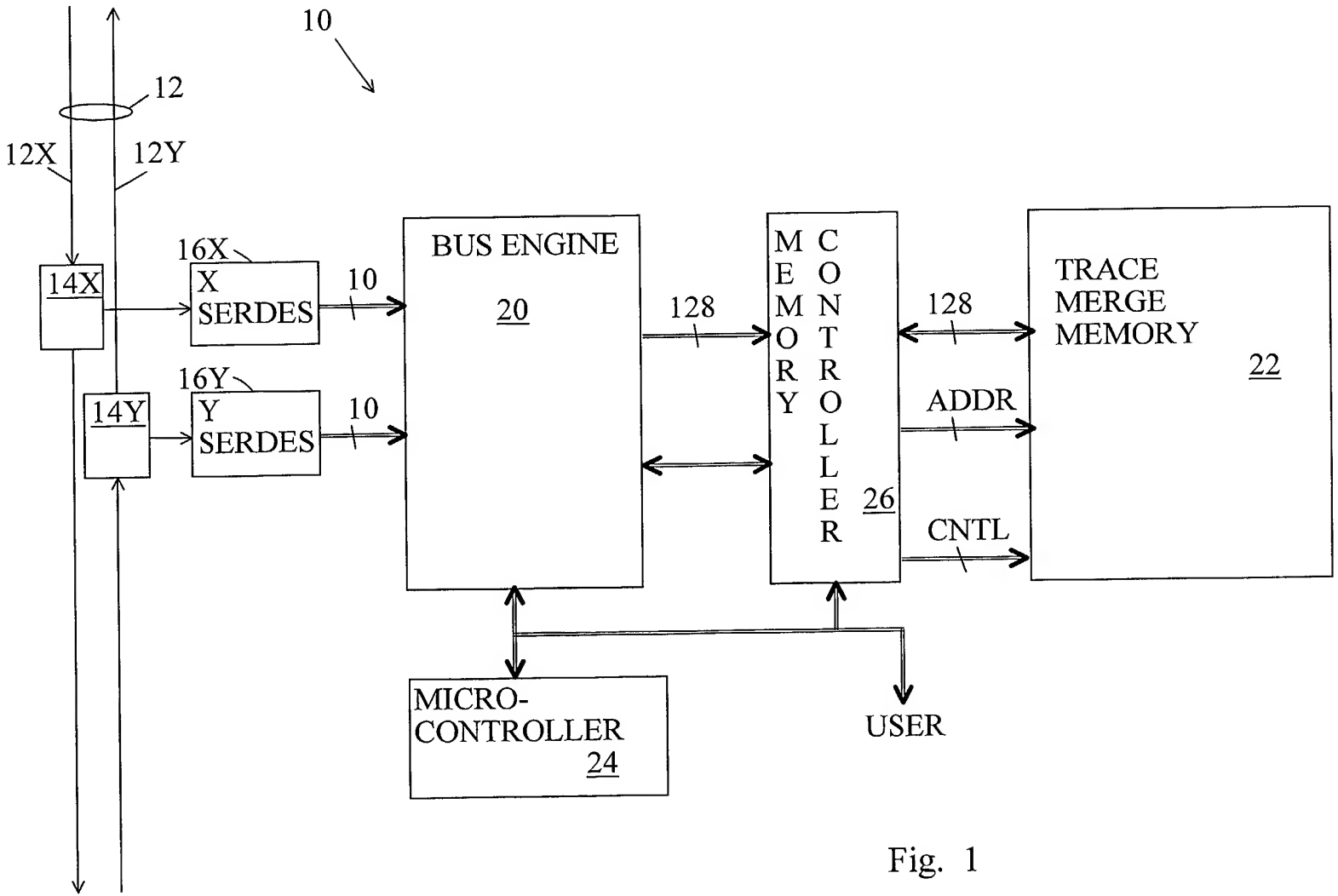
10
12
12X
12Y
14X
14Y
16X
X
SERDES
16Y
Y
SERDES
10
10
BUS ENGINE
20
128
MEMORY CONTROLLER
26
128
ADDR
CNTL
TRACE
MERGE
MEMORY
22
MICRO-
CONTROLLER
24
USER

Fig. 1

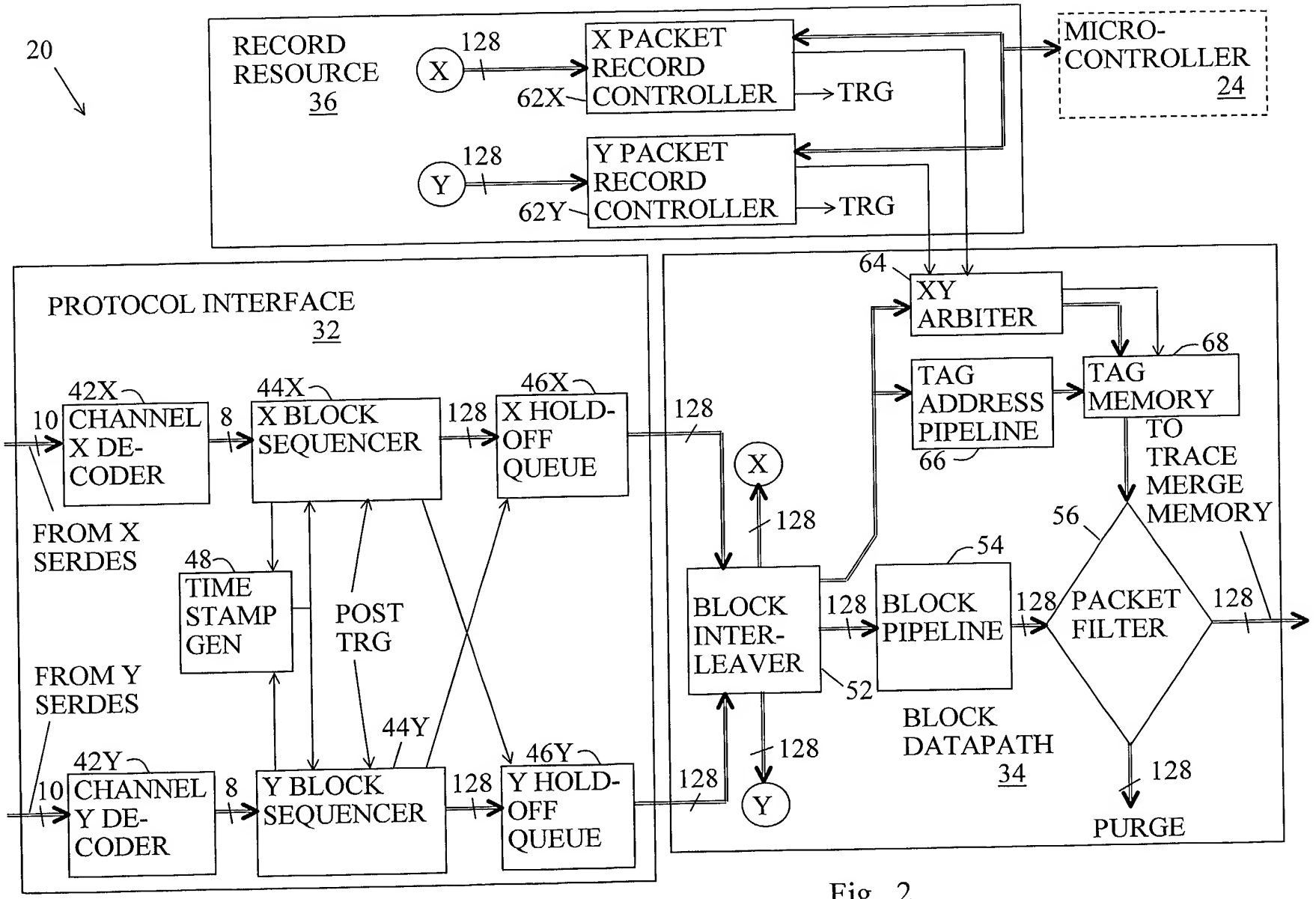
20
RECORD RESOURCE 36
X
128
X PACKET RECORD CONTROLLER
62X
TRG
Y
128
Y PACKET RECORD CONTROLLER
62Y
TRG
MICRO-CONTROLLER 24
PROTOCOL INTERFACE 32
64
XY ARBITER
68
TAG MEMORY
TAG ADDRESS PIPELINE
66
TO TRACE MERGE MEMORY
42X
10
CHANNEL X DE-CODER
8
44X
X BLOCK SEQUENCER
128
46X
X HOLD-OFF QUEUE
128
FROM X SERDES
48
TIME STAMP GEN
POST TRG
FROM Y SERDES
42Y
10
CHANNEL Y DE-CODER
8
44Y
Y BLOCK SEQUENCER
128
46Y
Y HOLD-OFF QUEUE
128
X
128
BLOCK INTER-LEAVER
128
BLOCK PIPELINE
54
52
128
Y
128
BLOCK DATAPATH 34
56
PACKET FILTER
128
128
PURGE

Fig. 2

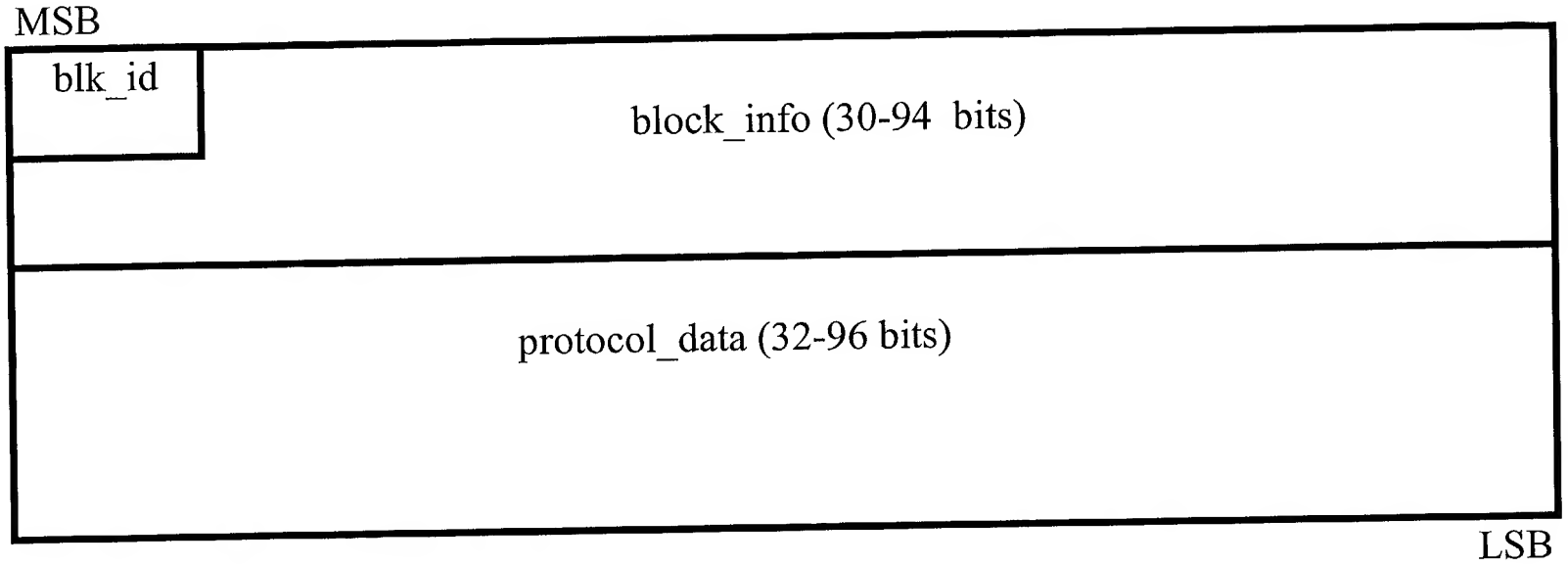

Fig. 3A

Code (bin)	Label (blk_id)	Comment
11	BLOCK_ONE	Data bytes in this block are the first in a packet, following a valid SDP or SLP control symbol.
10	BLOCK _MIDDLE	Data bytes in this block are the middle of a packet. Zero or more of this Block may appear contiguously, to accommodate the packet size.
01	BLOCK _LAST	Data bytes in this block are the last in a packet, and were followed by a valid EGP or EBP control symbol.
00	Undefined	

Fig. 3B

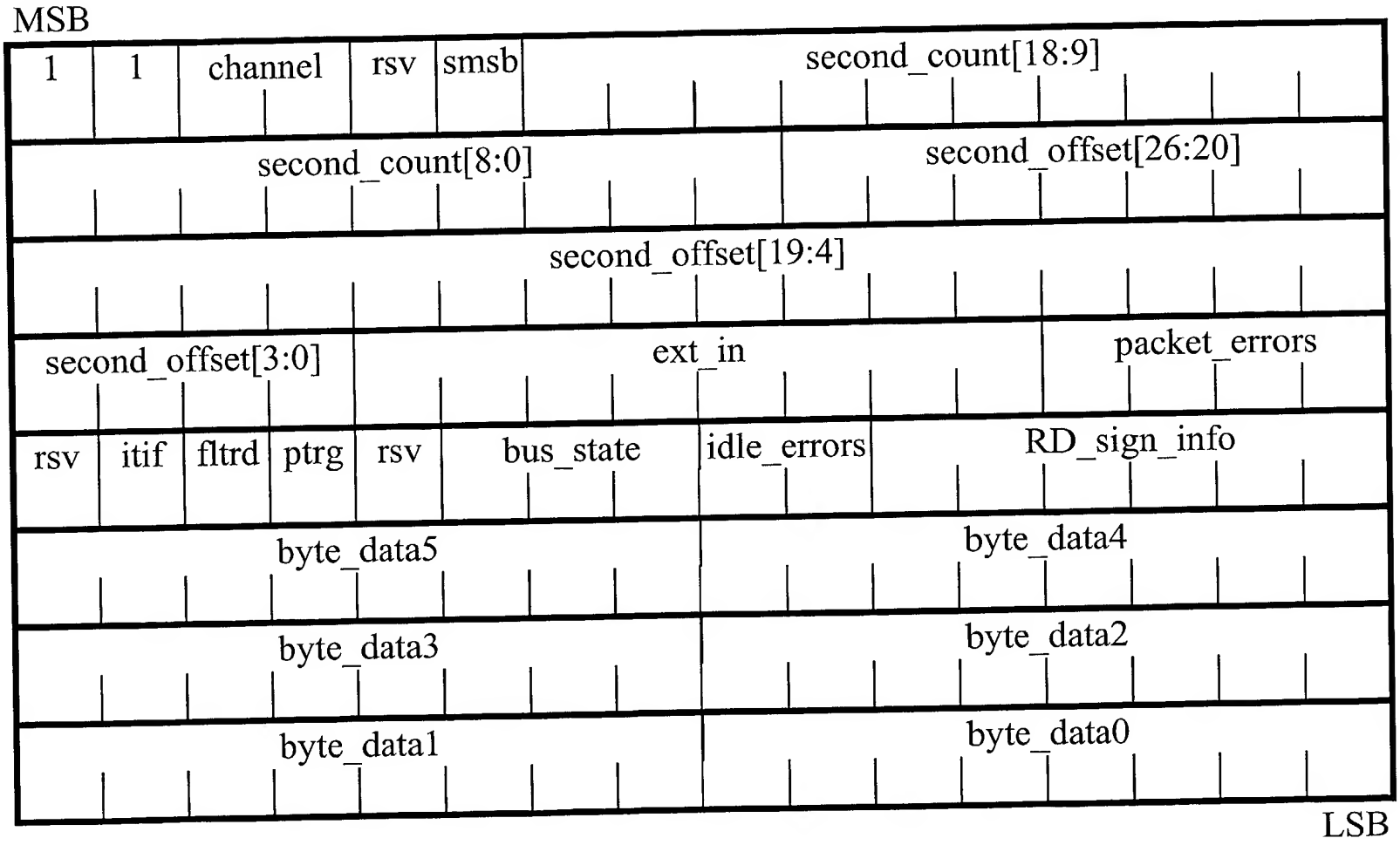
MSB
1
1
channel
rsv
smsb
second_count[18:9]
second_count[8:0]
second_offset[26:20]
second_offset[19:4]
second_offset[3:0]
ext_in
packet_errors
rsv
itif
fltrd
ptrg
rsv
bus_state
idle_errors
RD_sign_info
byte_data5
byte_data4
byte_data3
byte_data2
byte_data1
byte_data0
LSB

Fig. 3C

Bit Num	Label	Definition
3	END_BAD	When set this bit signifies this packet has had some error in it, notified by a EBP symbol that has arrived at the packet's end. When cleared the packet is good, notified by a EGP symbol that has arrived at the packet's end.
2	DELIMITER_ERROR	When set indicates a sequence error in the packet delimiters SDP, SLP, EGP, and EBP. The start delimiters SDP & SLP must be followed by an end delimiter EGP or EBP. This error indicates two start delimiters were received casing a sequence error.
1	SYMBOL_ERROR	When set indicates an error has been encountered in symbol integrity. This can happen if any 10-bit symbol received is not one of the legal 10-bit encoding for the type of data expected. Once this error is encountered, no more bytes will be written to recording memory until a new, valid SDP or SLP is received. Symbol error in this Block means the data in this Block is not valid.
0	ALIGNMENT_ERROR	When set, indicates an error has been encountered in the 16-bit data alignment. While IB packets must be 32-bit aligned only 16-bit word alignment is check by hardware. This is because packet delimiters are discarded and recorded packets normally end not 32-bit aligned.

Fig. 3D

Code (bin)	Label	Comment
000	DATA_PACKET	Determined by the SDP control symbol received before the packet
001	LINK_PACKET	Determined by the SLP control symbol received before the packet
010	ORDERED_SET	Defined as a COM control symbol followed by one of the valid bytes for a Training Sequence Ordered-Set (will not include any SKIP Ordered-Set)
011	Reserved	Reserved for future use
100	DISCONNECT	Link disconnect event, no data is recorded
101	CONNECT	Link connect event, no data is recorded
110	Reserved	Reserved for future use
111	Undefined	

Fig. 3E

Code (bin)	Label (most severe first)	Definition
01	SYMBOL_ERROR	When set indicates an error has been encountered in symbol integrity. This can happen if any 10-bit symbol received is not one of the legal 10-bit encoding for the type of data expected.
10	DISPARITY_ERROR	When set indicates a running disparity error occurred in the idle codes.
11	SKIP_ERROR	When set indicates a skip ordered set error occurred in the idle between packets. A skip sequence has either not been received on time (too early or too late), or an incorrect length skip sequence has been received.
00	NO_ERROR	No error in the idle between packets.

Fig. 3F

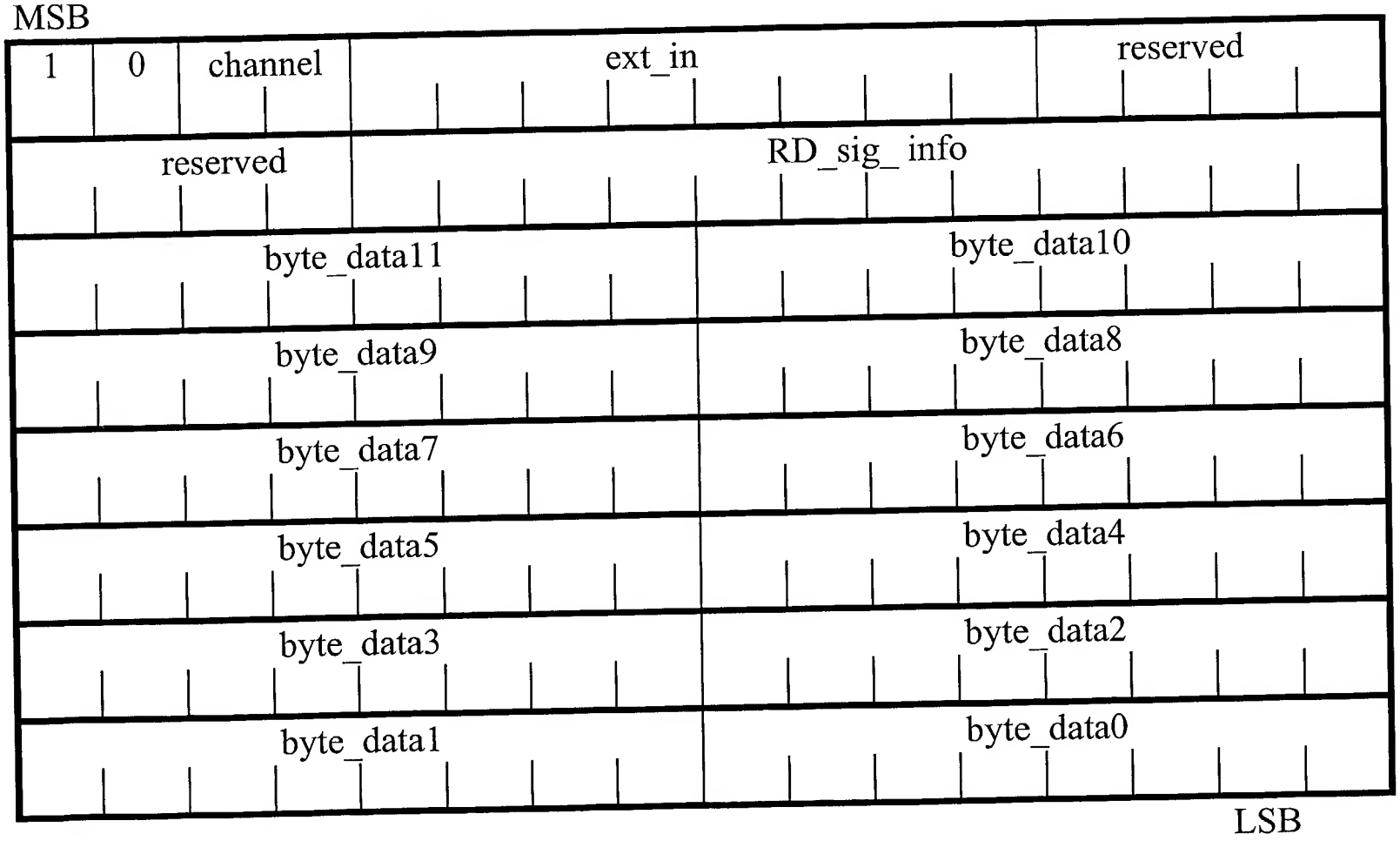
MSB
1
0
channel
ext_in
reserved
reserved
RD_sig_ info
byte_data11
byte_data10
byte_data9
byte_data8
byte_data7
byte_data6
byte_data5
byte_data4
byte_data3
byte_data2
byte_data1
byte_data0
LSB

Fig. 3G

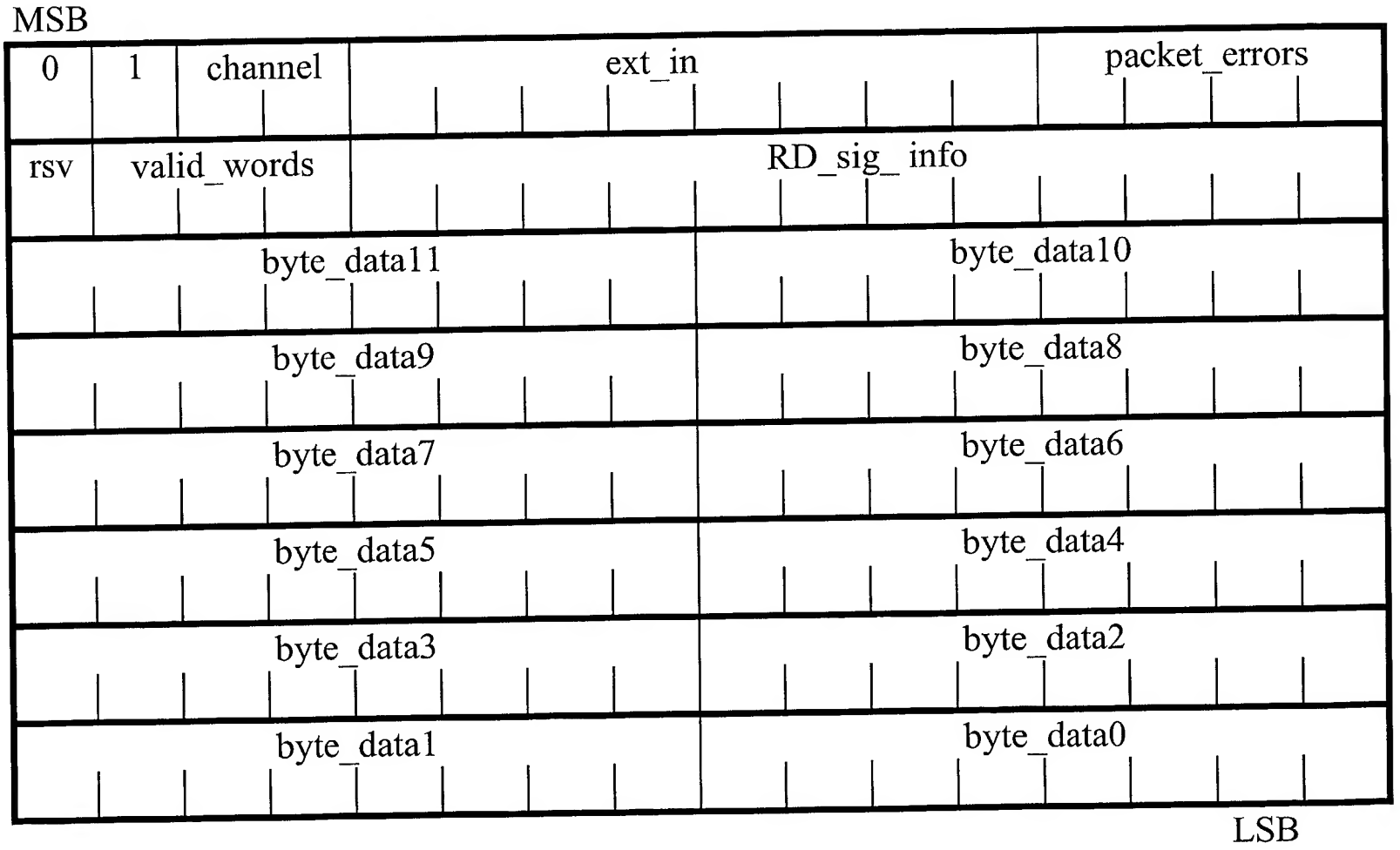
MSB
0
1
channel
ext_in
packet_errors
rsv
valid_words
RD_sig_ info
byte_data11
byte_data10
byte_data9
byte_data8
byte_data7
byte_data6
byte_data5
byte_data4
byte_data3
byte_data2
byte_data1
byte_data0
LSB

Fig. 3H

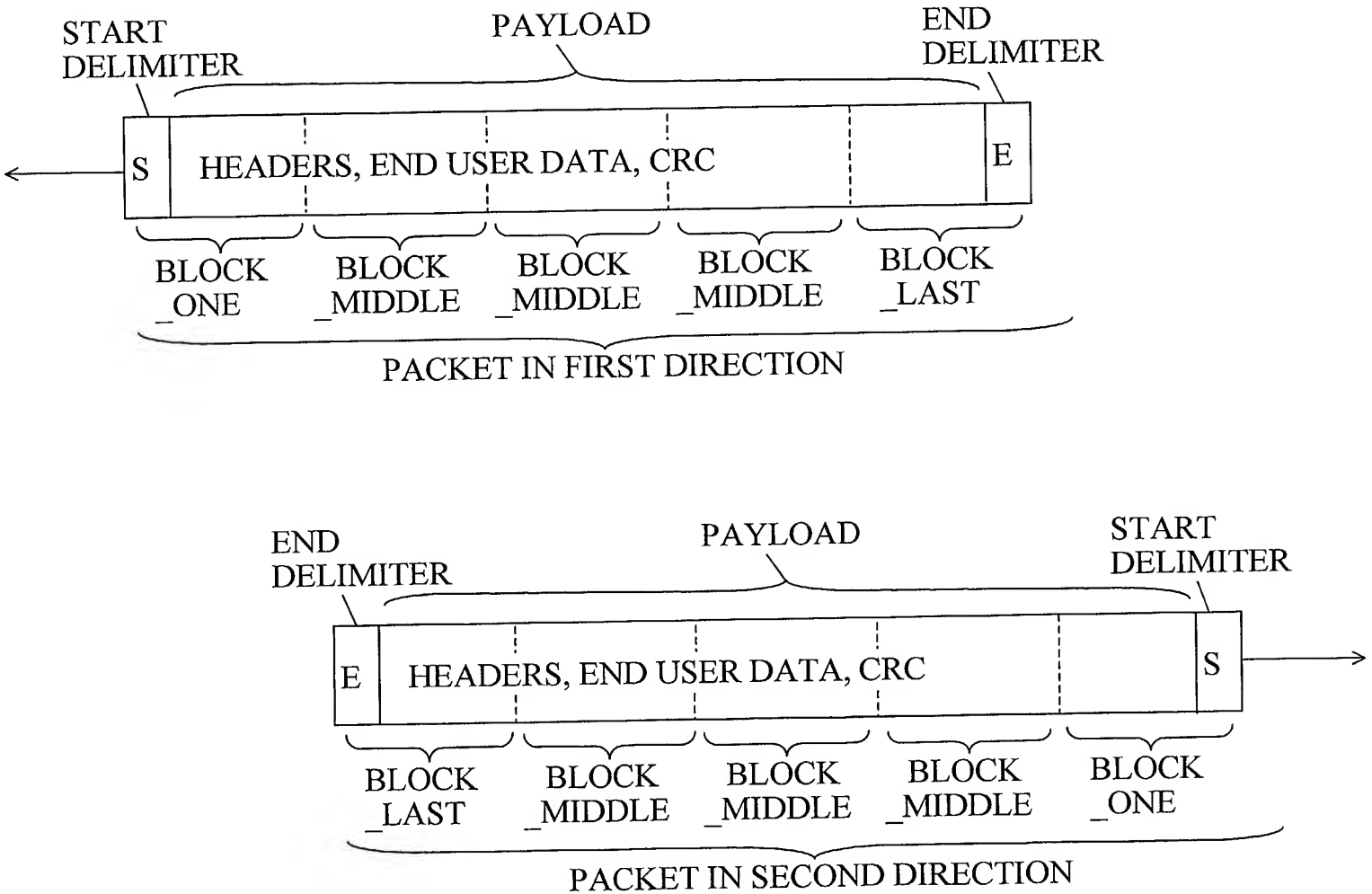

Fig. 4

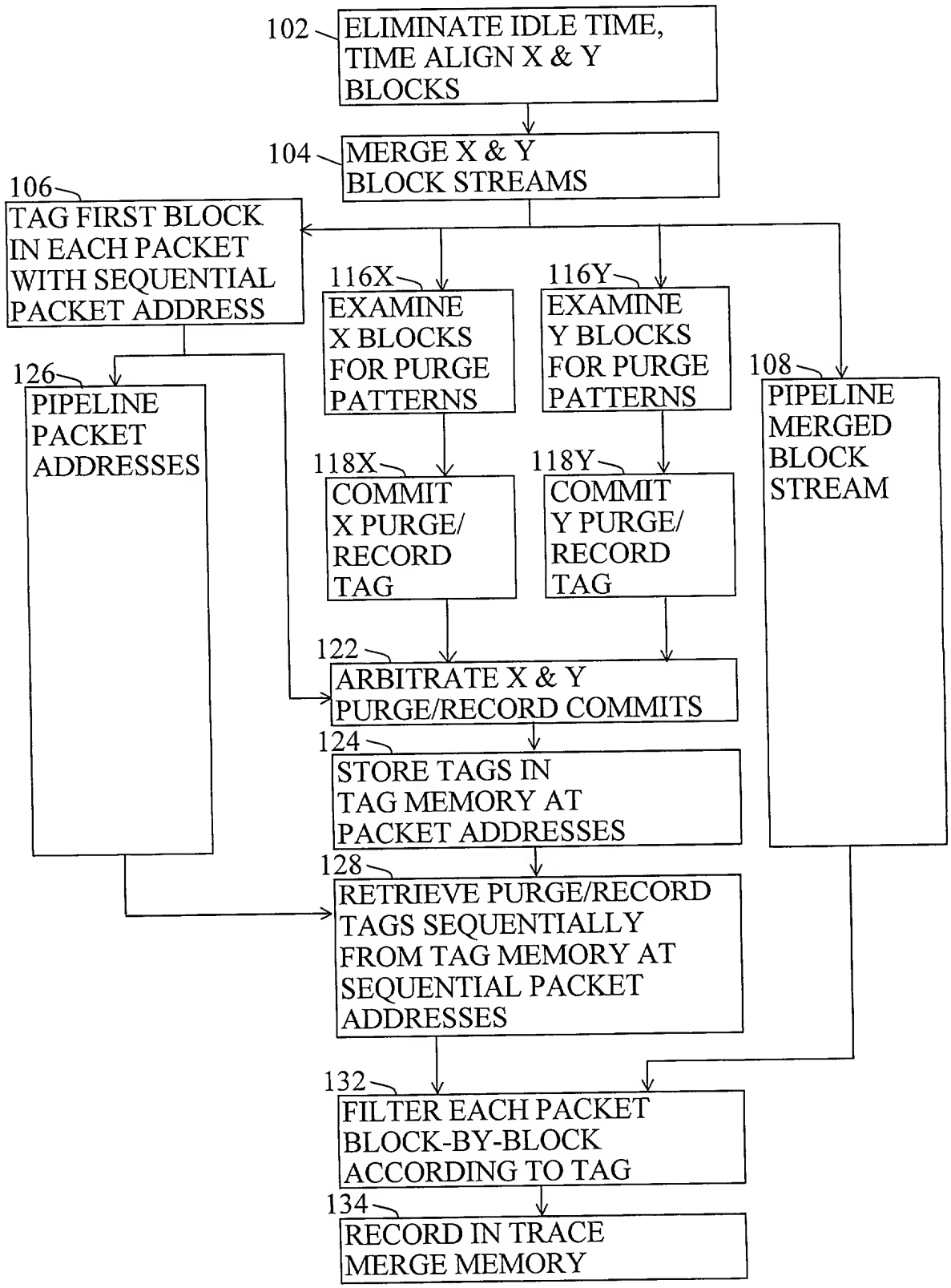
102
ELIMINATE IDLE TIME, TIME ALIGN X & Y BLOCKS
104
MERGE X & Y BLOCK STREAMS
106
TAG FIRST BLOCK IN EACH PACKET WITH SEQUENTIAL PACKET ADDRESS
116X
EXAMINE X BLOCKS FOR PURGE PATTERNS
116Y
EXAMINE Y BLOCKS FOR PURGE PATTERNS
108
PIPELINE MERGED BLOCK STREAM
126
PIPELINE PACKET ADDRESSES
118X
COMMIT X PURGE/ RECORD TAG
118Y
COMMIT Y PURGE/ RECORD TAG
122
ARBITRATE X & Y PURGE/RECORD COMMITS
124
STORE TAGS IN TAG MEMORY AT PACKET ADDRESSES
128
RETRIEVE PURGE/RECORD TAGS SEQUENTIALLY FROM TAG MEMORY AT SEQUENTIAL PACKET ADDRESSES
132
FILTER EACH PACKET BLOCK-BY-BLOCK ACCORDING TO TAG
134
RECORD IN TRACE MERGE MEMORY

Fig. 5

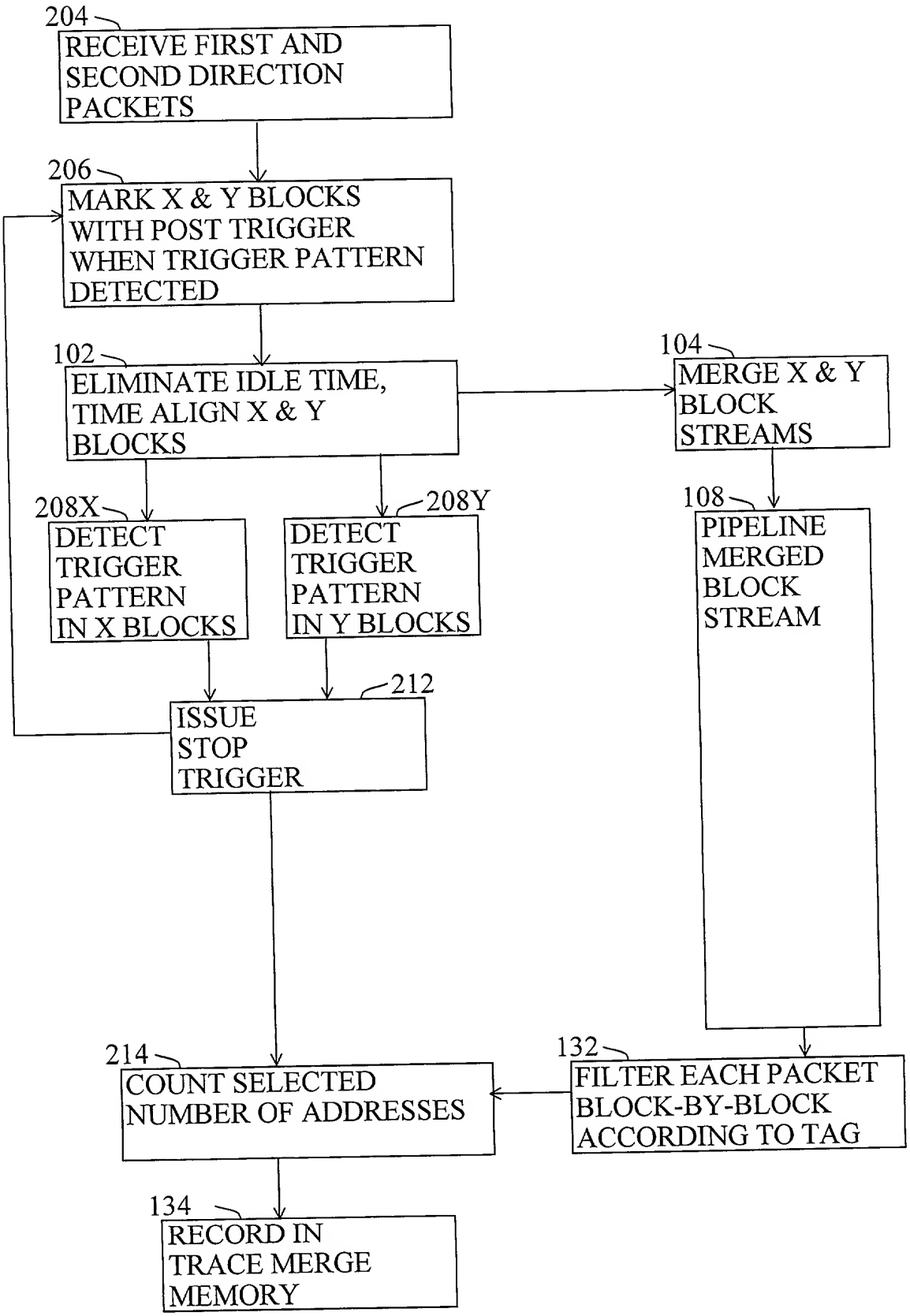
204
RECEIVE FIRST AND SECOND DIRECTION PACKETS
206
MARK X & Y BLOCKS WITH POST TRIGGER WHEN TRIGGER PATTERN DETECTED
102
ELIMINATE IDLE TIME, TIME ALIGN X & Y BLOCKS
104
MERGE X & Y BLOCK STREAMS
208X
DETECT TRIGGER PATTERN IN X BLOCKS
208Y
DETECT TRIGGER PATTERN IN Y BLOCKS
108
PIPELINE MERGED BLOCK STREAM
212
ISSUE STOP TRIGGER
214
COUNT SELECTED NUMBER OF ADDRESSES
132
FILTER EACH PACKET BLOCK-BY-BLOCK ACCORDING TO TAG
134
RECORD IN TRACE MERGE MEMORY

Fig. 6